



Merging BI Publisher PDF Documents with PeopleCode

PT 8.58

Randall Groncki

Combining BI Publisher PDF Reports

In some cases, it may be best to combine two existing PDF reports or split an existing PDF report into two separate documents for population, then recombine them before delivery.

Good examples are government forms with dependent information. Where the number of dependents exceeds the available positions on the form, the person would just attach an additional copy of that form page to the completed document to report all dependents. Separating the report into person information and dependent information, creating as many dependent pages as required and then recombining the two reports into one for delivery is a method to accomplish the task.

BI Publisher delivers a PDF Join utility to PeopleCode through an internal Java class. The code example below combines two PDF reports into one report.

```
import PSXP_ENGINE:*;  
&oMerger = create PSXP_ENGINE:PDFMerger();  
  
&PDF_Merge_Error = "";  
&PDF_Merge_Array = CreateArray(&FirstReportFilePath);  
&PDF_Merge_Array.Push(&SecondReportFilePath);  
  
&PDF_Merge_Result = &oMerger.mergePDFs(&PDF_Merge_Array, &PDF_Merge_File_name,  
&PDF_Merge_Error);
```

See Appendix A for a complete version of this PeopleCode event.

*Note: In order for the PSXP_ENGINE:PDFMerger class to merge PDFs, the “pdf-security” property on the all merged report definitions must be set to “False”

Definition
Template
Output
Properties
Security
Bursting

Report Name: X_PDF_FIRST

Report Properties

Property Group PDF Security

Property Settings

Property	Prompt	Password	Default
pdf-security	False		True
pdf-open-password			
pdf-permissions-password			
pdf-encryption-level			2

Adding Page Numbers

The PDF Merger class also contains a property object (PageNumber) to inject page numbers onto the combined result file. The convenience of this is that the page numbers are added after the multiple documents are merged

```
import PSXP_ENGINE:*;

Local PSXP_ENGINE:PDFMerger &oMerger = create PSXP_ENGINE:PDFMerger();
Local PSXP_ENGINE:PageNumber &oPageNumber = create PSXP_ENGINE:PageNumber();
Local PSXP_ENGINE:Watermark &oWatermark = create PSXP_ENGINE:Watermark();

&oPageNumber.FontName = "Helvetica";
&oPageNumber.FontSize = 10;
&oPageNumber.PositionX = 300;
&oPageNumber.PositionY = 20;
&oMerger.PageNumber = &oPageNumber;
```

PeopleBooks Documentation:

Refer to the PeopleTools PeopleBook PeopleCode API Reference for details on the PDF merging. The PDFMerge class is one of the delivered PI Publisher Classes.

https://docs.oracle.com/cd/F30998_01/pt858pbr2/eng/pt/tpcr/task_PDFMergerClass-647c06.html?pli=ul_d40e279_tpcr

Appendix A

Combining PDF Files

```
/* **** */
/* ** PeopleTools Tech Tips ** */
/* ** Randy Groncki 2020-12-10 ** */
/* ** peopletoolstechtips@gmail.com ** */
/* ** BI Publisher ** */
/* ** BI Pub PDF Join Example ** */
/* **** */
import PSXP_RPTDEFNMANAGER.*;
import PSXP_ENGINE.*;

Declare Function GetDirSeparator PeopleCode PSXPFUNCLIB.FUNCLIB FieldFormula;

Local XmlDoc &First_XMLDoc, &Second_XMLDoc;
Local XmlNode &childNode, &rowNode;
Local string &Xml_String, &FIRST_Filename_path, &FirstReportFilePath, &SecondReportFilePath;
Local string &PDF_Merge_File_name, &FIRST_XML_Filename_path, &Second_XML_Filename_path;
Local string &FirstReportFileName, &SecondReportFileName;
Local string &sDirSep, &PDF_Merge_Error;
Local File &FIRST_XML_File, &SECOND_XML_File, &PDF_Merge_File;

Local boolean &PDF_Merge_Result;
Local array of string &PDF_Merge_Array = CreateArrayRept("", 0);

Local Field &Fld_FilePath = GetRecord().FILE_PATH_NAME;
Local PSXP_RPTDEFNMANAGER:ReportDefn &oXML_PUB_1, &oXML_PUB_2;

/* Create FIRST XMLDoc */
&First_XMLDoc = CreateXmlDoc("<?xml version='1.0'?><root/>");
&rowNode = &First_XMLDoc.DocumentElement.AddElement("FIRST_Report");
&childNode = &rowNode.AddElement("FIRST_DATA");
&childNode.NodeValue = "FIRST DATA";
&Xml_String = &First_XMLDoc.GenFormattedXmlString();

&FIRST_XML_File = GetFile("FIRST_DATA.xml", "W", "UTF8");
&FIRST_XML_File.WriteLine(&Xml_String);
&FIRST_XML_Filename_path = &FIRST_XML_File.Name;
&FIRST_XML_File.Close();

/* Create the first pdf doc */
&oXML_PUB_1 = create PSXP_RPTDEFNMANAGER:ReportDefn("X_PDF_FIRST");
&oXML_PUB_1.Get();
&oXML_PUB_1.SetRuntimeDataXMLFile(&FIRST_XML_Filename_path);
&oXML_PUB_1.ProcessReport("", "", %Date, "");
CommitWork();

rem &oXML_PUB_1.DisplayOutput();

&sDirSep = GetDirSeparator();
&FirstReportFileName = &oXML_PUB_1.ID | "." | Lower(&oXML_PUB_1.GetOutDestFormatString(2));
&FirstReportFilePath = &oXML_PUB_1.OutDestination | &sDirSep | "RptInst" | &sDirSep | &FirstReportFileName;
```

Continued...

```
/* Create & Second XMLDoc */
&Second_XMLDoc = CreateXmlDoc("<?xml version='1.0'?><root/>");
&rowNode = &Second_XMLDoc.DocumentElement.AddElement("Second_Report");
&childNode = &rowNode.AddElement("Second_DATA");
&childNode.NodeValue = "Second DATA";
&Xml_String = &Second_XMLDoc.GenFormattedXmlString();

&SECOND_XML_File = GetFile("Second_DATA.xml", "W", "UTF8");
&SECOND_XML_File.WriteLine(&Xml_String);
&Second_XML_Filename_path = &SECOND_XML_File.Name;
&SECOND_XML_File.Close();

/* Create the second pdf doc */
&oXML_PUB_2 = create PSXP_RPTDEFNMANAGER:ReportDefn("X_PDF_SECOND");
&oXML_PUB_2.Get();
&oXML_PUB_2.SetRuntimeDataXMLFile(&Second_XML_Filename_path);
&oXML_PUB_2.ProcessReport("", "", %Date, "");
CommitWork();
/* display our individual PDFs that were merged */
rem &oXML_PUB_2.DisplayOutput();

&SecondReportFileName = &oXML_PUB_2.ID | "." | Lower(&oXML_PUB_2.GetOutDestFormatString(2));
&SecondReportFilePath = &oXML_PUB_2.OutDestination | &sDirSep | "RptInst" | &sDirSep | &SecondReportFileName;

/* Merge Documents */
/* create the objects */
Local PSXP_ENGINE:PDFMerger &oMerger = create PSXP_ENGINE:PDFMerger();
Local PSXP_ENGINE:PageNumber &oPageNumber = create PSXP_ENGINE:PageNumber();

/* define result file location and name */
&PDF_Merge_File = GetFile("PDF_MERGE.pdf", "W");
&PDF_Merge_File_name = &PDF_Merge_File.Name;
&PDF_Merge_File.Close();

/* set the page number */
rem &oPageNumber.FontName = "ZapfDingbats";
&oPageNumber.FontName = "Helvetica";
&oPageNumber.FontSize = 10;
&oPageNumber.PositionX = 300;
&oPageNumber.PositionY = 20;
&oPageNumber.StartNum = 2;
&oMerger.PageNumber = &oPageNumber;

/* load the pdfs to be merged into an array */
&PDF_Merge_Array.Push(&FirstReportFilePath);
&PDF_Merge_Array.Push(&SecondReportFilePath);

&PDF_Merge_Error = "";
&PDF_Merge_Result = &oMerger.mergePDFs(&PDF_Merge_Array, &PDF_Merge_File_name, &PDF_Merge_Error);

/* use the attachment functions to pop the new merged document to a new window */
&Returncode = PutAttachment("record://X_FILEATTACH", "PDF_MERGE.pdf", &PDF_Merge_File_name);
&Returncode = ViewAttachment("record://X_FILEATTACH", "PDF_MERGE.pdf", "PDF_MERGE.pdf");

&Fld_FilePath.Value = &PDF_Merge_File_name;

/* delete XML Data work files */
&FIRST_XML_File = GetFile("FIRST_DATA.xml", "W");
&FIRST_XML_File.Delete();
&SECOND_XML_File = GetFile("Second_DATA.xml", "W");
&SECOND_XML_File.Delete();
```